```
PROCEDURE PEEK (A : ADRESSE; VAR B : BYTE);
   TYPE MISCHTYP = (* siehe oben *);
   VAR X : MISCHTYP;
   BEGIN
   X.AD := A (* ADRESSE BESETZEN *);
   B := X.PT↑(* WERT HOLEN        *)
   END (* PEEK *);
```

Wer PEEK lieber als Funktion hat, muß schreiben:

```
FUNCTION PEEK (A : ADRESSE) : BYTE;
   TYPE MISCHTYP = (* siehe oben *);
   VAR X : MISCHTYP;
   BEGIN
   X.AD := A;
   PEEK := X.PT↑
   END (* PEEK *);
```

Ein paar Einschränkungen sind jedoch zu beachten. Hat der Compiler die Möglichkeit, per Option eine Laufzeitüberwachung ein- oder auszuschalten, dann muß diese unbedingt ausgeschaltet werden (Option T- oder R-). Des weiteren sind die Adressen über 32767 als negative Integerzahlen (Komplement!) einzugeben. Dabei entspricht −32768 der Adresse FFFFH und −1 der Adresse 8000H.
Leider gibt es einige Firmen, die beim Herstellen ihres Compilers nicht an maschinennahe Anwendungen gedacht haben und die Recordvarianten nicht übereinanderlegen. Bei diesen Compilern funktioniert die geschilderte Methode nicht.
Erprobt wurde das Verfahren auf der Pascal-Maschine von Western Digital und auf dem Großrechner CDC Cyber 175.
Ich würde mich freuen, Erfahrungsberichte über die Verwendung des Verfahrens auf Apple, TRS-80, PET und anderen zu erhalten.
Abschließend noch eine nützliche und praktisch erprobte Anwendung. Es wird hier ein Text bildschirmweise kopiert und per Tastendruck weitergeblättert.

```
BEGIN (* HAUPTPROGRAMM *)
ZEILE :=O;
WHILE NOT EOF(INPUT) DO
   BEGIN (* DATEI KOPIEREN *)
   ZEILE := ZEILE + 1;
   COPYLINE(INPUT,OUTPUT);
   IF ZEILE MOD 20 = O THEN
      (* WARTE BIS EINE TASTE GEDRUECKT WURDE *)
      REPEAT
         PEEK(TASTENPUFFER,DUMMY)
      UNTIL DUMMY   O;
   END;
END.
```

APPEND für TRS-80 L II

Um ein Programmteil an ein im Speicher vorhandenes Programm anzufügen, genügt es, das folgende Basicprogramm hinter das vorhandene Programm einzugeben und zu starten. Es belegt den Inhalt des Programmanfangszeigers in hex 40A4,5 mit der letzten Speicheradresse des aktuellen Programms. Diese läßt sich aus dem Inhalt des Variablenspeicherzeigers 40F9, A H berechnen, indem davon 2 subtrahiert wird.

```
A = PEEK (16633) : B = PEEK
(16634) : A = A-2
IF A < 0 THEN A = A + 256 : B = B-1
POKE 16548, A : POKE 16549, B : END.
```

Betrachtet man sich nun seinen Programmspeicher z. B. mit dem LIST-Befehl, so scheint er leer zu sein. Er ist jedoch nur verdeckt und ein Unterprogramm kann mit CLOAD zugeladen werden. Zu beachten ist dabei nur, daß die Zeilennummern des zugeladenen Programms größer sind, als die des verdeckten Programms. Dies ist bereits bei der Erstellung von Unterprogrammen dadurch zu berücksichtigen, daß diese mit genügend großen Zeilennummern beginnen.
Um das verdeckte Programm wieder zu aktivieren, muß in den Programmanfangszeiger wieder die Adresse 42E9 H eingegeben werden. Dies bewerkstelligt die Befehlsfolge:

```
POKE 16548,233 : POKE 16549,66.
```

Abschließend sind noch die drei Basiczeilen des Zuladeprogramms zu löschen, sofern sie nicht im Befehlsmodus eingegeben wurden. Werner Linsler

Berichtigung: Das Innenleben des MZ-80K

mc 1981, Heft 4, Seite 39

Die Bildunterschrift muß richtig lauten: „... B bezeichnet ein Byte, wobei jedes Bit als S für Null und L für Eins gesendet wird, das höchstwertige unüblicherweise zuerst. Jedem Byte ist ein L als Startbit vorangestellt." Im Bild ist die Anfangsimpulsfolge korrekterweise so: 22000 × S, 40 × L, 40 × S beim Deskriptor. 11000 × S, 20 × L, 20 × S beim Datensegment.